

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN

Facultad de Producción y Servicios

Escuela Profesional de Ingeniería de Sistemas



Estructuras de Datos y Algoritmos

Actividad: Fibonacci con algoritmos

Docente: Karim Guevara Puente de la Vega

Integrantes:

- Ticona Hareth, Anthony Joaquín**
- Umasi Cariapaza, Carlos Daniel**

Arequipa-Perú

2022

El tiempo de las pruebas realizadas fueron calculadas con “System.nanoTime()”. Cabe resaltar que el resultado obtenido por Fibonacci con números tan grandes, aun con variables long, no fue posible mostrarlo correctamente. Los valores que se muestran no son precisos, el programa no funciona correctamente con números tan extensos. En este informe únicamente se presentará la relación de los tiempos con respecto a la cantidad ingresada en cada algoritmo.

Algoritmo recursivo ($O(2^n)$):

```
//Carlos Daniel Umasi Cariapaza
import java.util.Scanner;
public class FibonacciRecursivo{
    public static int Fibonacci(int n){
        if(n < 2)
            return n;
        else
            return Fibonacci(n-1) + Fibonacci(n-2);
    }
    Run | Debug
    public static void main(String[]args){
        int num,numFib;
        Scanner scan = new Scanner(System.in);
        System.out.println(x: "Ingrese el número de Fibonacci que desea saber
        num = scan.nextInt();
        numFib = Fibonacci(num);
        System.out.println("El número de Fibonacci es: "+numFib);
    }
}
```

Tiempos (Aproximado):

Ingrese el número de Fibonacci que desea saber (Solo enteros positivos): 20
Su número de Fibonacci es: 6765
El cálculo fue realizado en un tiempo de: 532101 nanosegundos

0.5321 milisegundos

Ingrese el número de Fibonacci que desea saber (Solo enteros positivos): 30
Su número de Fibonacci es: 832040
El cálculo fue realizado en un tiempo de: 7126700 nanosegundos

7.1267 milisegundos

Ingrese el número de Fibonacci que desea saber (Solo enteros positivos): 40
Su número de Fibonacci es: 102334155
El cálculo fue realizado en un tiempo de: 504779799 nanosegundos

504.77 milisegundos = 0.5 segundos

Ingrese el número de Fibonacci que desea saber (Solo enteros positivos): 50
Su número de Fibonacci es: -298632863
El cálculo fue realizado en un tiempo de: 65034607100 nanosegundos

65034 milisegundos = 65 segundos

Algoritmo iterativo ($O(n)$):

```
//Anthony Joaquín Ticona Hareth
import java.util.Scanner;

public class FibonacciIterativo {

    public static void main(final String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el lugar del número Fibonacci que
desea saber: ");
        int numero = sc.nextInt();
        System.out.println(Fibonacci(numero));
    }

    public static long Fibonacci(int n) {
        long x = 1;
        long fibonacci = 0;
        for(long i = 0; i < n; i++) {
            fibonacci = fibonacci + x;
            x = fibonacci - x;
        }
        return fibonacci;
    }
}
```

Tiempos (Aproximado):

Ingrese el lugar del número Fibonacci que desea saber: 100
8736710778780434371

El cálculo fue realizado en un tiempo de 67800 nanosegundos

0.0678 Milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 1000
817770325994397771

El cálculo fue realizado en un tiempo de 89100 nanosegundos

0.0891 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 10000
-2872092127636481573

El cálculo fue realizado en un tiempo de 312000 nanosegundos

0.312 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 100000
2754320626097736315

El cálculo fue realizado en un tiempo de 1552000 nanosegundos

1.552 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 1000000
-4249520595888827205

El cálculo fue realizado en un tiempo de 2716200 nanosegundos

2.7162 milisegundos

Algoritmo de Memorización no iterativo ($O(n)$):

```
//Carlos Daniel Umasi Cariapaza
import java.util.Scanner;
public class FibonacciMemorización {
    public static int NumFibo(int n) {
        int arr[] = new int[n];
        for(int i = 2 ; i < n ; i ++ ) {
            arr[i] = -1;
        }
        arr[0] = 0;
        arr[1] = 1;
        int resp = FibMem(n-1,arr);
        return resp;
    }
    public static int FibMem(int n, int arr[]) {
        if(arr[n] == -1) {
            arr[n] = FibMem(n-1,arr) + FibMem(n-2,arr);
            return arr[n];
        }
        else
            return arr[n];
    }
}
Run | Debug
public static void main(String []args) {
    Scanner sc = new Scanner(System.in);
    int num,numF;
    System.out.println(x: "Ingrese el número fibonacci que desee saber");
    num = sc.nextInt();
    numF = NumFibo(num+1);
    System.out.println("Su número de Fibonacci es: "+numF);
}
```

Tiempos (Aproximado):

Ingrese el número fibonacci que desee saber(Debe ser entero positivo): 100
Su número de Fibonacci es: 3736710778780434371
El cálculo fue realizado en un tiempo de: 205801 nanosegundos

0.2058 milisegundos

Ingrese el número fibonacci que desee saber(Debe ser entero positivo): 1000
Su número de Fibonacci es: 817770325994397771
El cálculo fue realizado en un tiempo de: 346501 nanosegundos

0.3465 milisegundos

Ingrese el número fibonacci que desee saber(Debe ser entero positivo): 4000
Su número de Fibonacci es: 483944808890094715
El cálculo fue realizado en un tiempo de: 926400 nanosegundos

0.9264 milisegundos

Ingrese el número fibonacci que desee saber(Debe ser entero positivo): 6000
Su número de Fibonacci es: -7276231722249092800
El cálculo fue realizado en un tiempo de: 1039300 nanosegundos

1.0393 milisegundos

Ingrese el número fibonacci que desee saber(Debe ser entero positivo): 8000
Su número de Fibonacci es: -5778370123062214075
El cálculo fue realizado en un tiempo de: 1363401 nanosegundos

1.3634 milisegundos

Algoritmo Logarítmico ($O(\log(n))$):

```
//Anthony Joaquín Ticona Hareth
import java.util.Scanner;

public class FibonacciLogaritmico {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Ingrese el lugar del número Fibonacci que
desea saber: ");
        int numero = sc.nextInt();
        System.out.println(Fibonacci(numero));
    }

    public static long Fibonacci(int n) {
        //Matriz de Fibonacci
        //[1 1]
        //[1 0]
        //Al elevar n veces la matriz nos da el (n)ésimo número
Fibonacci
        //[fn+1 fn]^n
```

```

//[fn fn-1]

//Para valores iniciales
if(n <= 0){
    return 0;
}
if(n <= 2){
    return 1;
}
if(n == 3){
    return 2;
}

//Primera matriz
long a = 1;
long b = 1;
long c = 1;
long d = 0;
//Segunda matriz
long a1 = 1;
long b1 = 0;
long c1 = 0;
long d1 = 1;
//Valores reemplazables
long a2 = 1;
long a3 = 1;
long b3 = 1;
long c2 = 0;
long c3 = 1;
int numero = n-1;
while(numero > 0) {
    if(numero %2 != 0){
        a1 = a1*a +b1*c;
        b1 = a2*b + b1*d;
        c1 = c1*a + d1*c;
        d1 = c2*b + d1*d;
        //a2 y c2 reemplazan el valor inicial de la segunda
matriz
        a2 = a1;
        c2 = c1;
    }
    a = a*a+b*c;
    b = a3*b+b*d;

```

```

        c = c*a3+d*c;
        d = c3*b3+d*d;
        //a3 y b3 reemplazan el valor inicial de la matriz de
fibonacci
        a3 = a;
        b3 = b;
        numero = numero/2;
    }
    return a1;
}
}

```

Tiempos (Aproximado):

Ingrese el lugar del número Fibonacci que desea saber: 100
 329515843675406567

El cálculo fue realizado en un tiempo de 68100 nanosegundos

0.0681 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 1000
 -6835392897037256341

El cálculo fue realizado en un tiempo de 75900 nanosegundos

0.0759 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 10000
 -4917687938149703899

El cálculo fue realizado en un tiempo de 68300 nanosegundos

0.0683 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 100000
 8239071430176001369

El cálculo fue realizado en un tiempo de 68800 nanosegundos

0.0688 milisegundos

Ingrese el lugar del número Fibonacci que desea saber: 1000000
 -3458583702150674943

El cálculo fue realizado en un tiempo de 71800 nanosegundos

0.0718 milisegundos